REMARKS

This application has been carefully reviewed in light of the Office Action of November 22, 2006, wherein:

A. Claims 1-4 and 20-22 were rejected under 35 USC 102(e) as being anticipated by US Patent No. 6,327,622 to Jindal et al..

Claim Amendments

Claims 1-4 and 20-22 are in the application. Independent claims, Claims 1 and 20 are amended herein to make explicit that which was already implicit in the claims. In particular, considering Claim 1, that claim has been amended to require that "the persistence policy, if applicable, overrides the load balancing policy by controlling the allocation." The same amendments have been made to independent Claim 20.

This amendment is supported by page 3, lines 10-16, of the SUMMARY OF THE INVENTION section of the specification, and page 25, lines 6-15 of the DETAILED DESCRIPTION section of the specification, which sections respectively states as follows:

> A first embodiment of the invention comprises a system for applying a persistence policy to **override** allocation of a resource based on application of a load balancing policy. The system comprises first logic for determining if a persistence policy is applicable to a service request, and if so, allocating a resource request to the request based on application of a load balancing policy if the persistence policy is determined to be inapplicable as determined by the first logic. (emphasis added)
>
> * * *
>
> In one implementation, a history table 130 is included within the first logic 126 for maintaining a list of allocations which exist or which recently expired. When a service request is received, logic 128 within first logic 126 accesses the history table 130 in accordance with the persistence policy 129 to determine if an allocation exists or recently expired for the originator of the service request. If an allocation exists or recently expired for the originator of the service request, logic 131 **overrides** any allocation of the resource by second logic 127 based on application of a load balancing policy, and directs, through signal line 135, allocation of the resource to the request originator which resource was identified through application of the persistence policy 129. (emphasis added)

A. 35 USC 102(e)

In the Office Action, the Examiner rejected Claims 1-4 and 20-22 under 35 USC 102(e) as being anticipated by U.S. Patent No. 6,327,622 to Jindal et al., hereinafter the "Jindal patent."

Claims 1 and 20

The amendment to Claims 1 and 20 is being made to make explicit what was implicit in the claims, which is a key distinction of the claimed invention over the prior art. This distinction is the ability of the persistence policy to override the load balancing policy. As the BACKGROUND OF THE INVENTION section explains, efficiency is achieved by the persistence policy overriding the load balancing policy:

> Prior approaches to server load balancing typically emphasize avoiding overloading of any particular server, but fail to adequately consider other factors, such as whether a connection already exists or existed in the recent past for the client which has issued a service request. If such a connection exists or existed in the recent past, it may be more efficient to re-allocate the same server to the client in order to handle the current service request rather than allocate a different server based on application of a server load balancing policy. Page 3, lines 1-7.

In response to the Applicants' arguments on page 2, regarding Claims 1 and 20, the Examiner stated "Jindal clearly teaches wherein the special object structure is constructed to apply the selected load balancing policy only if ... it is determined the instructions or executable code for the specialized object that will determine a preferred server does not already exist (See col. 12, lines 12-23). Furthermore, Applicant is reminded that although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims."

As noted in section 706.02 of the MPEP, for anticipation under 35 USC 102, the reference must teach every aspect of the claimed invention. The Applicants submit that the Jindal patent does not teach, disclose or suggest each and every element of Claims 1 or 20.

Claim 1 claims, in part, "first logic in the network for determining if a persistence policy is applicable to a service request, ..." and "second logic in the network for allocating the resource based on a load balancing policy... if the persistence policy is determined to be

inapplicable as determined by the first logic, wherein the persistence policy, if applicable, overrides the load balancing policy by controlling the allocation."

The Examiner references col. 12, lines 13-23 to rebut the Applicants' argument that the lookup table of the Jindal patent changed depending upon the load-balancing policy on a periodic or regular basis. The Applicants submit that col. 12, lines 13-23 of the Jindal patent does not show that "wherein the special object structure is constructed to apply the selected load balancing policy only if it is determined the instructions or executable code for the specialized object that will determine a preferred swerver does not already exist," as stated by the Examiner.

First, the Applicants respectfully submit that the Jindal patent states that:

> when client 120 attempts to connect to application 104, the access request is received by central server100. Central server 100, through lookup table 102, identifies a preferred server offering an instance of program 104 and routes the client request accordingly. The server identified in lookup table 102 may be determined according to a load-balancing policy .... Further, the server identified in the lookup table 102 is updated or changed from time to time in accordance with the selected policy in order to distribute client requests among the instances of the application. (Jindal, Col. 6, lines 35-45).

Second, col. 12, lines 13-23 cited by the Examiner refer to the states shown in Figure 4. Figure 4 "is a flow chart demonstrating the generation of objects in a load-balancing framework in accordance with an embodiment of the present invention." (see col. 4, lines 1-3) The Applicants respectfully point to Fig. 5 which "is a flow chart demonstrating the registration of objects within the load balancing framework and their use in monitoring an instance of a load-balancing application in accordance with an embodiment of the present invention." (see col. 4, lines 4-7) As explained in col. 12, lines 26-37

> With reference now to Fig. 5, illustrative registration and monitoring stages of the illustrated method are depicted. For present purposes, the term registration refers to the registration of individual objects ... within a load balancing framework, including their creation ... from the object structures ... produced in the generation state depicted in Fig. 4. ***In the monitoring stage, information is collected for the purpose of identifying a preferred server in accordance with a selected load balancing policy.*** (emphasis added)

Thus, as explained through the citations of the Jindal patent, Figure 4 outlines the generation of objects, while Figure 5 outlines the monitoring of the objects generated in Figure 4. Thus, as stated in col. 14, lines 24-26 "[i]n state 528 the specialized object updates the lookup

table for the central server to indicate the preferred server." Thus, the Applicants submit that the Jindal patent does not teach, disclose, or suggest, "wherein the persistence policy, if applicable, overrides the load balancing policy by controlling the allocation," as is claimed in Claims 1 and 20.

Further, The Jindal patent is directed toward "[a] method ... for providing load balancing requests for an application among a plurality of instances of the application operating on a plurality of servers. A policy is selected for choosing a preferred server from the plurality of servers according to a specified status or operational characteristic of the application instances." See Abstract of Jindal. Further, the Jindal patent discloses that

> information concerning instances of an application ... operating on multiple computer servers is collected and analyzed to identify a "preferred" server. Illustratively, a preferred server is the server to which client requests for the application are to be routed for processing. A preferred server is identified on a regular or periodic basis, and may be the same or different from the server previously identified. By periodically changing the preferred server, client requests are load-balanced between the participating servers.

(Jindal, Col. 4, lines 57-67).

Additionally, the Jindal patent states that

> when client 120 attempts to connect to application 104, the access request is received by central server100. Central server 100, through lookup table 102, identifies a preferred server offering an instance of program 104 and routes the client request accordingly. The server identified in lookup table 102 may be determined according to a load-balancing policy .... Further, the server identified in the lookup table 102 is updated or changed from time to time in accordance with the selected policy in order to distribute client requests among the instances of the application.

(Jindal, Col. 6, lines 35-45).

The Applicants submit that the use of a lookup table in the Jindal patent is not the same thing as a persistence policy, which is claimed in Claim 1. As explained on page 21 of the present application, "[p]ersistence attempts to force the client request to the server that handled the last request from the same client." The lookup table of the Jindal patent merely stores the preferred server determined by the load-balancing policy. Additionally, the lookup table of the Jindal patent is changed depending upon the load-balancing policy on a periodic or regular basis. Therefore, the Applicants submit that the use of the lookup table in the Jindal patent is not a persistence policy as claimed in Claim 1.

Claim 20 claims, in part, "determining if a persistence policy is applicable to a service request, and, if so, allocating a resource to the request based on the persistence policy." For the reasons stated above, the Applicants submit that the Jindal patent does not teach, disclose, or suggest "a persistence policy." Since the Jindal patent does not disclose each and every element of Claim 20, the Applicants submit that the Jindal patent does not anticipate Claim 20.

Claim 2

Claim 2 is dependent upon Claim 1; therefore, Claim 2 is patentable at least due to its dependence upon an allowable base claim. Furthermore, Claim 2 is patentable over the Jindal patent.

Claim 2 claims, in part, "wherein the first logic determines if a persistence policy is applicable ... through consideration of whether or not an allocation exists or recently expired for the originator of the service request."

As mentioned above, the Applicants submit that the Jindal patent does not teach, disclose, or suggest "wherein, the persistence policy, if applicable, overrides the load balancing policy by controlling the allocation." Additionally, the Applicants submit that the Jindal patent does not teach, disclose, or suggest "wherein the first logic determines if a persistence policy is applicable ... through consideration of whether or not an allocation exists or recently expired for the originator of the service request," as is claimed in Claim 2.

In response to the Applicant's arguments on page 3, the Examiner stated "Jindal clearly teaches wherein the special object structure is constructed to apply the selected load balancing policy only if it is determined the instructions or executable code for the specialized object that will determine a preferred server does not already exist. (See col. 12, lines 13-23). The Applicants respectfully disagree with the Examiner's interpretation of the Jindal patent. The Applicants refer to their arguments regarding Claims 1 and 20, where the Applicants address the Examiner's position regarding Col. 12, lines 13-23.

Additionally, in rejecting Claim 2, the Examiner cited col. 11, lines 55-67 of the Jindal patent. Col. 11, lines 55-67 discusses "the existing load-balancing framework is examined to determine whether an IMO ... already exists for collecting data concerning an instance of the load-balanced application."

Col. 8, line 23 defines an IMO as an individual monitor object. "IMOs 210, 212, and 214 collect information from the status objects 200, 202, and 204 respectively," see col. 8, lines 27-29. "[C]onfiguration of the status objects (e.g., the data they collect) depends upon the policy that has been selected for choosing the preferred server," see col. 7, lines 59-60. "[I]nformation gathered by the application-specific status objects is used by other objects and/or modules in the load-balancing framework in order to determine a preferred server," see col. 5, lines 62-65. Therefore, the IMOs discussed in col. 11, lines 55-67 are used in the *load-balancing policy*. Thus, the Applicants submit that the cited portions of the Jidal patent do not teach, disclose, or suggest "wherein the first logic determines if a persistence policy is applicable ... through consideration of whether or not an allocation exists or recently expired for the originator of the service request," as is claimed in Claim 2 (emphasis added).

Therefore, the Applicants submit that Claim 2 is patentable over the Jindal patent.

Claims 3 and 21

In response to the Applicants' arguments on page 3, the Examiner stated:

As per claims 3 and 21, Jindal clearly teaches wherein the special object structure is constructed to apply the selected load balancing policy only if it is determined the instructions or executable code for the specialized object that will determine a preferred server does not already exist (see. Col. 12, lines 13-23, col. 9, lines 6-58).

The Applicants respectfully disagree with the Examiner on the interpretation of the Jindal patent. The Applicants refer to their arguments regarding Claims 1 and 20, where the Applicants address the Examiner's position regarding Col. 12, lines 13-23.

Further, the Applicants respectfully request that he Examiner clarify the response to the Applicants previous arguments. Previously, the Applicants submitted that the Jindal patent does not teach, disclose, or suggest "determining whether an allocation exists or recently expired for the originator of the resource request," as is claimed in Claim 3 (emphasis added). In the response to the Applicants' arguments, the Examiner did not address this portion of the Applicants response. If the Examiner continues to maintain the position that the Jindal patent anticipates Claims 3 and 21, the Applicants respectfully request that the Examiner address the Applicants arguments previously submitted, which are set forth below for convenience.

Regarding Claims 3 and 21, the Examiner stated the Jindal patent teaches "a system for allocating a resource to a resource request having an originator based on application of a persistence policy comprising: first logic for determining whether an allocation exists or recently expired for the originator of the resource request, and, if so, identifying the resource which is the subject of the existing or recently expired allocation (See col. 9, lines 6-58); and a second logic for allocating the resource, once identified, to the resource request (See 6, lines 35-45).

As noted in section 706.02 of the MPEP, for anticipation under 34 USC 102, the reference must teach every aspect of the claimed invention. The Applicants submit that the Jindal patent does not teach, disclose or suggest each and every element of Claims 3 or 21.

As previously stated, the Jindal patent does not teach, disclose, or suggest a "persistence policy." Further, the Jindal patent does not teach, disclose, or suggest "determining whether an allocation exists or recently expired for the originator of the resource request," as is claimed in Claim 3 (emphasis added). As explained in col. 9, lines 6-58, the Jindal patent discloses that "when load balancing is performed ... a status object gathers load and/or operational information for an instance of the application being load balanced. ... [D]ata collected ... is analyzed in accordance with the selected policy and a preferred server is identified. ... [U]pdater object 230 updates lookup table 102 after the collected information is analyzed and a preferred server is selected." The sections of the Jindal patent cited by the Examiner discuss how the preferred server is selected based on the characteristics for each instance of an application and how the client is routed to the preferred server via a lookup table. The Applicants find no mention in the portions of the Jidal patent cited by the Examiner of "determining whether an allocation exists or recently expired for the originator of the resource request," as is claimed in Claim 3 (emphasis added).

Therefore, the Applicants submit that the cited portions of the Jindal patent do not teach, disclose, or suggest "determining whether an allocation exists or recently expired for the originator of the resource request."

Claim 21 claims, in part, "determining whether an allocation exists or recently expired for the originator of the resource request, and, if so, identify the resource which is the subject of the existing or recently expired allocation; and allocating the resource, once identified, to the resource request."

As previously stated with respect to Claim 3, the Jindal patent does not teach, disclose, or suggest "determining whether an allocation exists or recently expired for the originator of the resource request." Thus, Claim 21 is patentable over the Jindal patent for the same reasons given above for Claim 3.

Claims 4 and 22

Claim 4 is dependent upon Claim 3 and Claim 22 is dependent upon Claim 21; therefore, Claims 4 and 22 are patentable at least due to their dependence upon an allowable base claim.

Conclusion

For all the foregoing reasons, reconsideration of and withdrawal of all outstanding rejections is respectfully requested. The Examiner is earnestly solicited to allow all claims, and pass this application to issuance.

To expedite allowance of this case, the Examiner is earnestly invited to call Robert C. Laurenson at (949) 759-5269.

Respectfully submitted,

Date: January 22, 2007

Viola T. Kung (Reg. No. 41,131)
Robert C. Laurenson (Reg. No. 34,206)

HOWREY LLP
2041 Fairview Park Drive, **Box No. 7**
Falls Church, VA 22042
FAX No. (703) 336-6950
Telephone No. (949) 759-5269